# (12) EUROPEAN PATENT APPLICATION

(84) Designated Contracting States:
AT BE CH CY DE DK ES FI FR GB GR IE IT LI LU MC NL PT SE
Designated Extension States:
AL LT LV MK RO SI

(30) Priority: 09.03.1998 US 36782

(71) Applicant: AT&T Corp.
New York, NY 10013-2412 (US)

(72) Inventors:
• Bharucha, Behram H.
Millburn, New Jersey 07041 (US)

• Bradley, James Federick
Middletown, New Jersey 07748 (US)
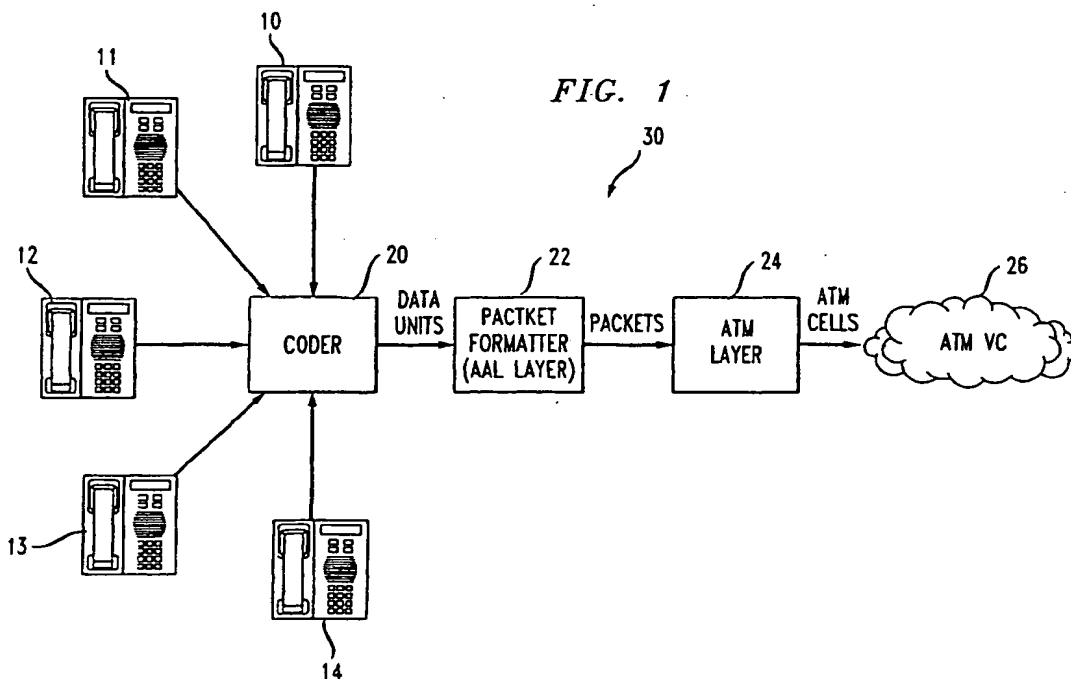• Chu, Thomas P.
Englishtown, New Jersey 07726 (US)

(74) Representative: Asquith, Julian Peter et al
Marks & Clerk, -
Nash Court,
Oxford Business Park South
Oxfordshire OX4 2RU (GB)

## (54) Method and apparatus for transmitting compressed packetized voice over ATM

(57) A system transmits compressed voice packets over ATM. The system receives a plurality of voice data units and an encoding rule for each voice data unit. A packet is formed from the received data units. The packet includes a header. The header includes the amount of data units in the packet, and the encoding rule for each packet. The packet is then placed in an ATM cell which is transmitted on an ATM virtual channel. The ATM cell includes at most one packet.

FIG. 1

EP 0 942 622 A2

## Description

### FIELD OF THE INVENTION

[0001] The present invention is directed to digital voice communications. More particularly, the present invention is directed to a method and apparatus for transmitting compressed packetized voice over asynchronous transfer mode.

### BACKGROUND OF THE INVENTION

[0002] Asynchronous Transfer Mode ("ATM") is based on the switching and multiplexing technique selected by the International Telecommunication Union ("ITU") for the broadband access for the Integrated Services Digital Network ("ISDN"). ATM utilizes a fixed-size packet, referred to as an ATM cell. Each ATM cell is 53 octets (one octet equals one byte) long and is divided into a 5 octet header and 48 octet information field.

[0003] The ITU has further defined an ATM adaptation layer ("AAL"). The AAL is the upper layer of the ATM layer and is responsible for converting user information from higher layers into a 48 octet payload field while the ATM cell layer adds the 5 octet header to make the 53 octet ATM cell. A family of AALs, AAL1 through AAL5, has been defined to handle diverse user applications. The AAL is usually subdivided into two parts: the Common Part Convergence Sub-layer ("CPCS") and the Service-Specific Convergence Sub-layer ("SSCS"). The SSCS defines formatting for different applications onto ATM cells.

[0004] One increasingly popular use of ATM is to transmit voice packets during telephone calls. In some networks, uncompressed voice packets are transmitted. A single uncompressed voice channel typically requires a bandwidth of 64 Kbps. However, various methods are used to compress voice in order to reduce the bandwidth requirements. For example, ITU G.728 voice coders typically compress voice to 16 Kbps. Of course, as the voice is more and more compressed, the reproduced voice quality is somewhat degraded.

[0005] When ATM is used to transmit voice telephone calls, multiple telephone calls are usually multiplexed onto a single physical link. For example, if 30 voice calls were multiplexed on a single physical link, and voice compression is used so that each voice call transmits at 16 Kbps, the total bandwidth required would be 480 Kbps (30 X 16 Kbps) plus overhead.

[0006] Besides compression. some telecommunication systems suppress silence during a telephone call in order to further reduce the bandwidth. In other words, if there is silence during a telephone conversation, the silence is not transmitted and the required bandwidth is temporarily reduced. When 30 voice calls are multiplexed together as described above, at any given moment approximately 15 voice calls require bandwidth while the other 15 voice calls consist of silence that can be suppressed. Thus, in implementing the 30 voice call system, only 240 Kbps (15 X 16 Kbps) may typically be required. However, if only 240 Kbps is allocated, a problem exists during periods when over 15 voice calls at once are in active conversation (i.e., no silence) and more than 240 Kbps of bandwidth is temporarily needed.

[0007] One known way to solve this problem is for the system to throw away voice packets (i.e., not transmit selected packets) during the periods when the increased bandwidth is required. However, this will cause the received voice quality to be severely degraded during those periods. Another way to solve the problem is to temporarily "down-speed" the amount of compression (i.e., the compression speed) during the periods when increased bandwidth is required. For example, some G.728 coders down-speed to 12.8 Kbps and 9.6 Kbps during periods of congestion. Using this method, the degradation of the received voice signal is minimized compared with dropping packets. When increased bandwidth is no longer required, the coder can "up-speed" back to its nominal speed of 16 Kbps.

[0008] One method for forming and transmitting voice packets that can accommodate down-speeding and up-speeding of voice compression has been proposed by the ITU. This method is referred to as the "I.trunk Transmitter Procedures for Voice Service" ("I.trunk"). I.trunk builds on AAL2 which is an AAL that supports variable size packets, sometimes referred to as "mini-packets." With I.trunk, whenever a compression speed changes, due to either down-speeding or up-speeding, the packet that is forming is cut off and a new packet is formed. Therefore, each packet is homogenous because each packet contains a single type of compression. Multiple AAL2 packets are included in one ATM cell.

[0009] One problem with I.trunk is that the mini-packets require two levels of switching for a large network. One layer of switching is the ATM cell layer. A second layer of switching is the AAL2 packet within the ATM cell layer, since I.trunk places multiple packets in each ATM cell. However, two layers of switching requires more complicated and more expensive equipment on an ATM network.

[0010] Based on the foregoing, there is a need for a method and apparatus for forming and transmitting voice packets over ATM cells that can accommodate down-speeding and up-speeding of voice compression without requiring "two-layer" switching and can enable the ATM cell to be efficiently packed.

### SUMMARY OF THE INVENTION

[0011] One embodiment of the present invention is a system for transmitting compressed voice packets over ATM. The system receives a plurality of voice data units and an encoding rule for each voice data unit. A packet is formed from the received data units. The packet includes a header. The header includes the amount of da-

ta units in the packet, and the encoding rule for each packet. The packet is then placed in an ATM cell which is transmitted on an ATM virtual channel. The ATM cell includes at most one packet. To achieve high packing efficiency, the ATM cell contains as many data units as possible, even when the data units are of different types.

**[0012]** The present invention allows a single packet to include data units formed with different encoding rules (i.e., a mixed mode packet). This accommodates the dynamic nature of up-speeding and down-speeding. Further, the present invention does not require two-layer switching because the ATM cell includes only one packet.

BRIEF DESCRIPTION OF THE DRAWINGS

**[0013]** Fig. 1 is a block diagram illustrating one embodiment of an ATM transmission system that includes the present invention.

**[0014]** Fig. 2 is a block diagram illustrating steps performed by a packet formatter in accordance with one embodiment of the present invention.

DETAILED DESCRIPTION

**[0015]** One embodiment of the present invention is a packet formatter that formats digitized speech data units into a single packet. The packet can include data units that were digitized using different encoding rules (i.e., different types of compression). A header is included that describes the encoding rule for each data unit. One packet and an ATM header comprise an ATM cell. No more than one packet can be included in each ATM cell, and the packet can be homogenous or mixed mode.

**[0016]** Fig. 1 is a block diagram illustrating one embodiment of an ATM transmission system 30 that includes the present invention. The system includes a plurality of telephones 10-14 coupled to a coder 20. Telephones 10-14 each transmit an analog voice signal during a telephone call.

**[0017]** The outputs of telephones 10-14 are received by coder 20 which then digitizes the received audio speech. Coder 20 also compresses the voice signals at different compression speeds depending on the amount of voice signals received and the amount of bandwidth available. In one embodiment, coder 20 is a G.728 coder that normally operates at 16 Kbps. However, coder 20 can down-speed in the event of source congestion to 12.8 Kbps and 9.6 Kbps. The digitized data output from coder 20 is in the form of data units. A data unit is based on a fixed time interval. For example, in one embodiment, each data unit consists of a 5 millisecond time interval of compressed voice. This results in the length of a voice data unit of 10, 8 and 6 octets when coder 20 is operating at 16 Kbps, 12.8 Kbps and 9.6 Kbps, respectively. Coder 20 can also output uncompressed speech using ITU G.711 64 Kbps PCM, or output a Silence Insertion Descriptor ("SID") when no outgoing speech is

transmitted during a telephone call. The different possible speech outputs of coder 20 (e.g., 64 Kbps, 16 Kbps, 12.8 Kbps, etc.) are referred to as the encoding rules of system 30.

**[0018]** Coder 20 is coupled to packet formatter 22. Packet formatter 22 is part of the AAL layer. Packet formatter 22 formats the received data units into packets that are placed in an ATM cell.

**[0019]** ATM layer 24 receives the packets from packet formatter 24. ATM layer 24 adds a 5 octet header to the packets to form ATM cells. The ATM cells are then transmitted over an ATM virtual channel 26.

**[0020]** Packet formatter 22 requires multiple parameters as inputs. The first parameters are the encoding rules. In one embodiment, the encoding rules are:

- G.711 64 Kbps PCM;
- G.728 16 Kbps LD-CELP;
- G.728 12.8 Kbps LD-CELP;
- G.728 9.6 Kbps LD-CELP; and
- Silence Insertion Descriptor (SID).

**[0021]** For each encoding rule, the following parameters are specified:

- the time interval for the data unit associated with the encoding rule (e.g., 5 milliseconds);
- the maximum number of data units in a homogenous packet; and
- the transfer format of a data unit.

**[0022]** In the present invention, packets generated by packet formatter 22 can be homogenous or mixed mode. A homogenous packet is a packet in which all data units were formed using the same encoding rule. The packet length of a homogenous packet can be determined by multiplying the number of data units by the length of a data unit.

**[0023]** A mixed mode packet is a packet which includes data units formed using different encoding rules. For example, in a packet with four units, one data unit may be formed using 16 Kbps, and three data units may be formed using 9.6 Kbps. For a mixed mode packet, the following parameters are specified:

- the maximum number of mixed mode data units that can fit into a packet; and
- bits patterns that identify the encoding rule used for each data unit.

**[0024]** It is possible that, in some instances, the maximum packet length is reached before the packet contains the maximum number of data units. Of course, when that occurs. the packet is formed and transmitted without the maximum number of data units.

**[0025]** Fig. 2 is a block diagram illustrating steps performed by packet formatter 22 in accordance with one embodiment of the present invention. The steps of Fig.

2 are executed for every data unit output by coder 20.

[0026] At step 100, a buffer within packet formatter 22 that holds the received data units is initialized. The buffer is initialized by emptying it so that it does not store any data units. Further, counters are initialized to zero. Counters include the number of data units, and the length of the packet.

[0027] At step 110, a new data unit is received from coder 20. Included with the data unit is an indication of the encoding rule of the data unit (e.g., 16 Kbps).

[0028] At step 120, formatter 22 determines if the "new" packet is homogenous. The new packet consists of the data units already in the buffer plus the new data unit received at step 110. If all of the data units of the new packet have the same encoding rule, then the new packet is homogenous. If all of the data units of the new packet do not have the same encoding rule, then the new packet is not homogenous (i.e., the new packet is mixed mode). Of course, when step 120 is executed for the first time, the new packet will always be homogenous because buffer is empty and the new packet includes only the new data unit.

[0029] If the new packet is homogenous, at step 130 the "new" number of data units is compared with the maximum allowable number of data units. The new number of data units consists of the number of data units already in the buffer plus the new data unit (i.e., the data units of the new packet). The maximum number of data units is a fixed parameter that is dependent on the encoding rule of the data units (all data units at step 130 should have the same encoding rule since they are homogenous).

[0030] If the new number of data units is not equal to the maximum number of new data units at step 130, the process branches to step 140. A step 140, the new data unit is appended to the data units already in the buffer and the process branches to step 110.

[0031] If the new number of data units is equal to the maximum number of data units at step 130, the process branches to step 180. At step 180, the new data unit is appended to the data units already in the buffer and a packet is formed. The packet includes the data units in the buffer and a header. Details on the header are described below. After step 180, the process branches to step 100.

[0032] Step 150 is executed if the new packet is not homogenous at step 120. At step 150, the "new" packet length is compared to the maximum packet length. The new packet length is the length of the new packet which includes the data units already in the buffer plus the new data unit. The maximum packet length, or payload, is a fixed parameter and is typically 48 octets minus a 2 or 3 octet header.

[0033] If the new packet length is equal to the maximum packet length at step 150, the process branches to step 180.

[0034] If the new packet length is greater than the maximum packet length at step 150, the process branches to step 190. At step 190, the packet is formed with the data units already in the buffer and a header. The packet does not include the new data unit. At step 200, the buffer is initialized (i.e., emptied) and the new data unit is then added to the buffer. The process then branches to step 110.

[0035] If the new packet length is less than the maximum packet length at step 150, the process branches to step 160. Step 160 is identical to step 130. If at step 160, the number of data units is greater than or equal to the maximum number of data units, the process branches to step 190. If at step 160, the number of data units is less than the maximum number of data units, the process branches to step 170. Step 170 is identical to step 140. After step 170, the process branches back to step 110.

[0036] The packet formed by packet formatter 22 includes a plurality of data units and a header. In one embodiment, the header includes at least two pieces of information: the number of data units in the packet, and the encoding rule for each data unit. The header can also include whether the packet is homogenous or mixed mode, the sequence number of the packet, and a CRC. Further, each data unit will be numbered and each packet typically will have a sequence number. The sequence number for a packet is the number of the first data unit in that packet. The sequence number is used by a receiver to detect missing packets and facilitate error recovery.

[0037] The following are embodiments of packets generated by the present invention. The first is an AAL2-based embodiment, the second is an AAL5-based embodiment, and the third embodiment is based on a new type of AAL.

## AAL2 Based Embodiment

[0038] In this embodiment, the basic time interval for the data units is 5 milliseconds. This results in the length of the data units of 10, 8 and 6 octets when coder 20 is operating at 16 Kbps, 12.8 Kbps and 9.6 Kbps, respectively. The silence insertion descriptor is assumed to be 1 octet long.

[0039] Each ATM cell contains a single AAL2 packet. Each cell will have 44 octets of usable payload when the one octet pointer field and three octet AAL2 header are subtracted out. Based on the lengths of the different data units, a cell can fit:

- 4 data units at 16 Kbps (4 X 10 = 40 octets);
- 5 data units at 12.8 Kbps (5 X 8 = 40 octets);
- 7 data units at 9.6 Kbps (7 X 6 = 42 octets).

[0040] Therefore, every time a homogenous packet is transmitted by packet formatter 22 with an encoding rule of 16 Kbps, the sequence number will increase by 4. The increment is 5 and 7 for an encoding rule of 12.8 Kbps and 9.6 Kbps, respectively.

[0041] Because the sequence number can be incremented by up to 7 for a packet. the sequence number modulus is selected to be 16, the maximum allowable by AAL2. Further, a type 2 packet. as defined by AAL2, is used. A type 2 packet has a 3 octet header in addition to the AAL2 header. The three octet header for mixed mode packets in one embodiment is encoded as follows:

[0042] Bits 1-2: A packet type identifier.

[0043] Bits 3-5: The number of data units in the packet, maximum of 5 data units.

[0044] Bits 6-15: Two bits are assigned to each data unit as indicated above from first to last. The pattern 00, 01, 10, and 11 indicates that the encoding rule for the data unit is 16 Kbps, 12.8 Kbps and 9.6 Kbps, and silence, respectively.

[0045] Bits 16-19: The sequence number.

[0046] Bits 20-24: CRC-5.

## AAL5 Based Embodiment

[0047] An AAL5 packet has an 8 octet trailer, leaving a payload of 40 octets. In this embodiment, with a single packet per ATM cell, and having a basic time interval of 5 milliseconds, a cell can fit:

- 4 data units at 16 Kbps (4 X 10 = 40 octets);
- 5 data units at 12.8 Kbps (5 X 8 = 40 octets);
- 6 data units at 9.6 Kbps (6 X 6 = 36 octets).

[0048] In the 8 octet AAL5 trailer, there is one octet of UUI that can be used by packet formatter 22. Bits 1-3 are used to identify the packet. For example, a "000" can be a homogenous PCM packet, a "001" can be a homogenous 16 Kbps packet, and a "100" can be a mixed mode packet. Each mixed mode packet in one embodiment of the present invention includes an additional two octet header formatted as follows:

[0049] Bits 14: The number of data units in the packet, maximum of 6 data units.

[0050] Bits 5-16: Two bits are assigned to each data unit as indicated above from first to last. The pattern 00, 01, 10, and 11 indicates that the encoding rule for the data unit is 16 Kbps, 12.8 Kbps and 9.6 Kbps, and silence, respectively.

## New AAL Embodiment

[0051] In this embodiment. AAL2 packets are modified to provide a more efficient packet. First, the start field that is always aligned with the beginning of the cell is deleted. The start field is not needed because in the present invention there is only one packet per ATM cell. The CID field in the AAL2 packet header is also not needed so it is converted to the following usage:

Bit 1: Indicates whether the packet is mixed mode or homogenous.

Bits 2-7: Sequence numbering.

[0052] If the mode indicator (bit 1 of the CID field) is set to 0, it indicates that the packet is not a mixed mode speech packet. The UUI field of the packet is then used to classify the packet as either a homogenous speech packet with the encoding rule specified, or as another packet type such as a control and signaling information packet (e.g., an AAL2 type 3 packet).

[0053] If the mode indicator is set to 1, it indicates that the packet is a mixed mode speech packet, and the UUI field is used to indicate the number of data units in the packet. The packet payload contains additional overhead which describes the encoding rule of the different data units in the packet. The size of the overhead depends on the maximum number of data units that can appear in a packet and the number of encoding rules that are supported.

[0054] In one embodiment, a 3 octet header is chosen. As there are 4 different encoding rules, 2 bits are assigned to specify each data unit as follows:

| | |
|----|-----------|
| 00 | silence |
| 01 | 16 Kbps |
| 10 | 12.8 Kbps |
| 11 | 9.6 Kbps. |

[0055] Again, the basic interval is 5 milliseconds. With a 3 octet header, the bit allocation scheme is as follows:

| Octet | Bit | Assignment |
|-------|-------|------------------------------|
| 1 | 1 - 2 | Encoding rule of 1st data unit |
| | 3 - 4 | Encoding rule of 2nd data unit |
| | 5 - 6 | Encoding rule of 3rd data unit |
| | 7 - 8 | Encoding rule of 4th data unit |
| 2 | 1 - 2 | Encoding rule of 5th data unit |
| | 3 - 4 | Encoding rule of 6th data unit |
| | 5 - 6 | Encoding rule of 7th data unit |
| | 7 - 8 | Encoding rule of 8th data unit |
| 3 | 1 - 2 | Encoding rule of 9th data unit |
| | 3 | Reserved |
| | 4 - 8 | CRC-5 |

[0056] As shown, this embodiment can support up to 9 data units in a mixed mode packet. In contrast, in the AAL2-based embodiment that uses a type 2 packet, only a maximum of 5 data units can be supported in a mixed mode packet.

[0057] As disclosed, the present invention transmits compressed voice by forming a mixed mode packet and transmitting one packet per ATM cell. The mixed mode packet allows for down-speeding and up-speeding of

the speech signals as well as efficient packing of an ATM cell. Further, limiting the ATM cell to a single packet eliminates the requirement of two-layer switching.

[0058] Several embodiments of the present invention are specifically illustrated and/or described herein. However, it will be appreciated that modifications and variations of the present invention are covered by the above teachings and are within the purview of the appended claims without departing from the spirit and intended scope of the invention.

[0059] For example, although each described embodiment utilized a G.728 coder, other types of coder can be used in the present invention. In other embodiments, a G.726 coder can be used to include the following encoding rules: G.711 64 Kbps PCM; G.726 32 Kbps AD-PCM; G.726 24 Kbps ADPCM; G.726 16 Kbps ADPCM; and Silence Insertion Descriptor (SID).

## Claims

1. A method of transmitting compressed voice packets over asynchronous transfer mode (ATM) comprising the steps of:

   receiving a plurality of voice data units and a plurality of encoding rules, wherein one encoding rule is received for each data unit;
   forming a packet comprising the data units and a header; and
   transmitting the packet within an ATM cell;

   wherein the header comprises:

   an amount of data units in the packet; and
   the encoding rule for each data unit.

2. The method of claim 1, wherein the voice data units and encoding rules are received one at a time, said method further comprising the step of:

   for each encoding rule received, determining whether the encoding rule is different from previously received encoding rules.

3. The method of claim 2, further comprising the step of:

   for each data unit received, determining whether a number of data units in a new packet equals a maximum number of data units;
   wherein the packet is formed when the number of data units in the new packet equals the maximum number of data units.

4. The method of claim 2, further comprising the step of

   for each data unit received, determining whether a length of a new packet is greater than or equal to a maximum packet length;
   wherein the packet is formed when the length of the new packet is greater than or equal to the maximum packet length.

5. The method of claim 1, wherein the ATM cell has at most one packet.

6. The method of claim 1, wherein the encoding rules comprise a plurality of speeds.

7. The method of claim 6, wherein said plurality of speeds comprise 16 Kbps voice compression, 12.8 Kbps voice compression and 9.6 Kbps voice compression.

8. The method of claim 6, wherein said plurality of speeds comprise 32 Kbps voice compression, 24 Kbps voice compression and 16 Kbps voice compression.

9. The method of claim 6, wherein the encoding rules further comprise silence.

10. The method of claim 1, wherein the packet is an AAL packet.

11. The method of claim 10, wherein the AAL packet is an AAL2 packet.

12. The method of claim 10, wherein the AAL packet is an AAL5 packet.

13. A system for transmitting compressed voice packets over asynchronous transfer mode (ATM) comprising:

   a coder that digitizes voice signals into a plurality of voice data units using a plurality of encoding rules;
   a packet formatter coupled to said coder that forms a packet from the plurality of data units; and
   an ATM layer coupled to said packet formatter that generates an ATM cell from said packet;

   wherein said packet comprises a header, and said header comprises:

   an amount of data units in said packet; and
   the encoding rule for each data unit.

14. The system of claim 13, wherein said coder is a G. 728 coder.

15. The system of claim 13, wherein said coder is a G.

726 coder.

16. The system of claim 13, wherein the ATM cell includes at most one packet.

17. The system of claim 14, wherein the encoding rules comprise 16 Kbps voice compression, 12.8 Kbps voice compression and 9.6 Kbps voice compression.

18. The system of claim 17, wherein the encoding rules further comprise silence and 64 Kbps voice.

19. The system of claim 13, wherein the packet is an AAL packet.

20. A method of forming an asynchronous transfer mode (ATM) cell, said method comprising the steps of:

    (a) receiving a voice signal;
    (b) determining an amount of available bandwidth;
    (c) forming a voice data unit from the voice signal in accordance with an encoding rule, wherein said encoding rule is dependent on the amount of available bandwidth;
    (d) receiving a plurality of voice data units by repeating steps (a) -(c);
    (e) forming a packet from the received data units, said packet comprising a header; and
    (f) forming the ATM cell with said packet;

wherein the header comprises an amount of data units in the packet and the encoding rule used to form each of the data units in the packet.

21. The method of claim 20, wherein the header further comprises a sequence number of the packet, and a type of packet.

22. The method of claim 21, wherein the type of packet comprises a homogenous packet and a mixed mode packet.

23. The method of claim 20, wherein the ATM cell includes at most one packet.

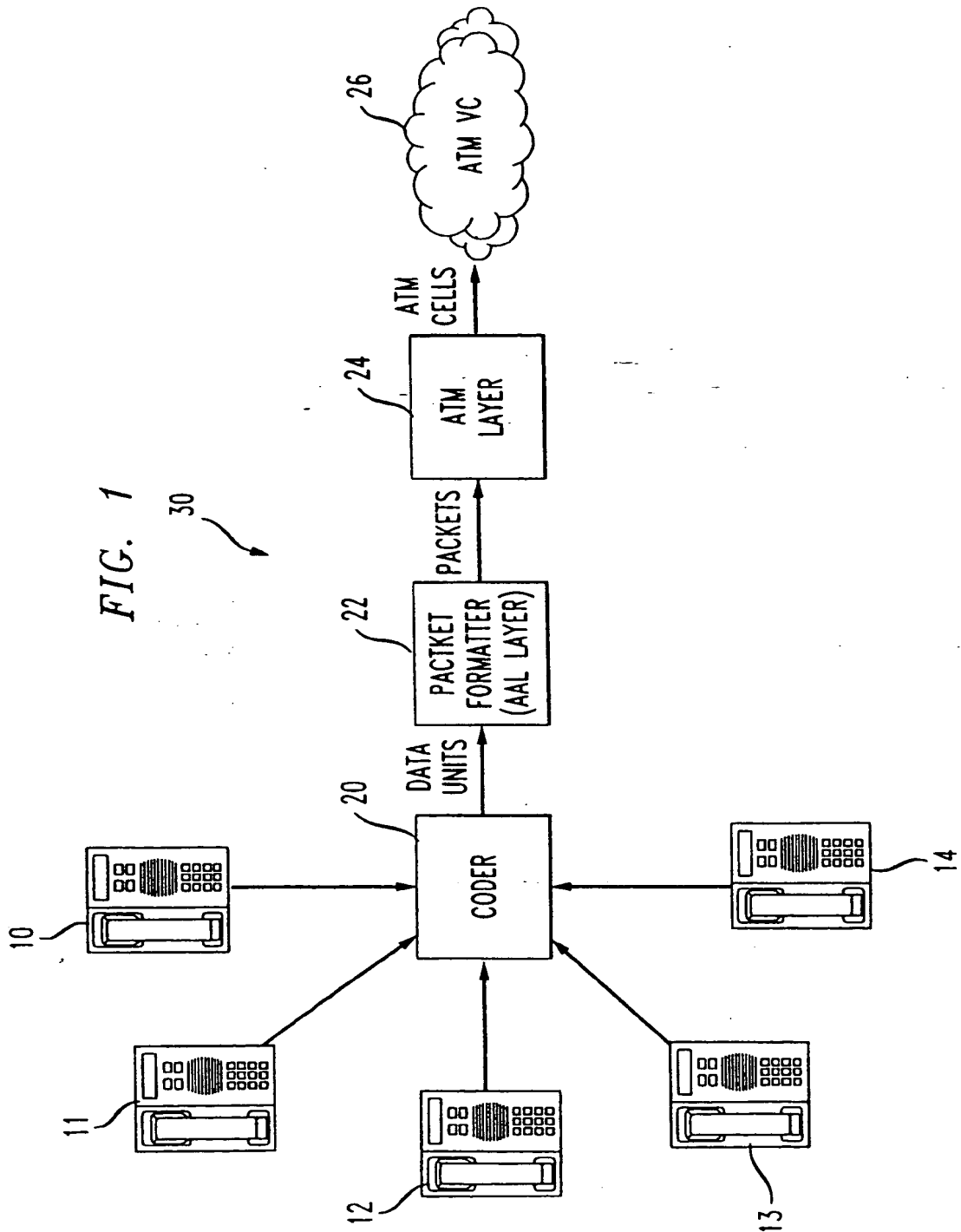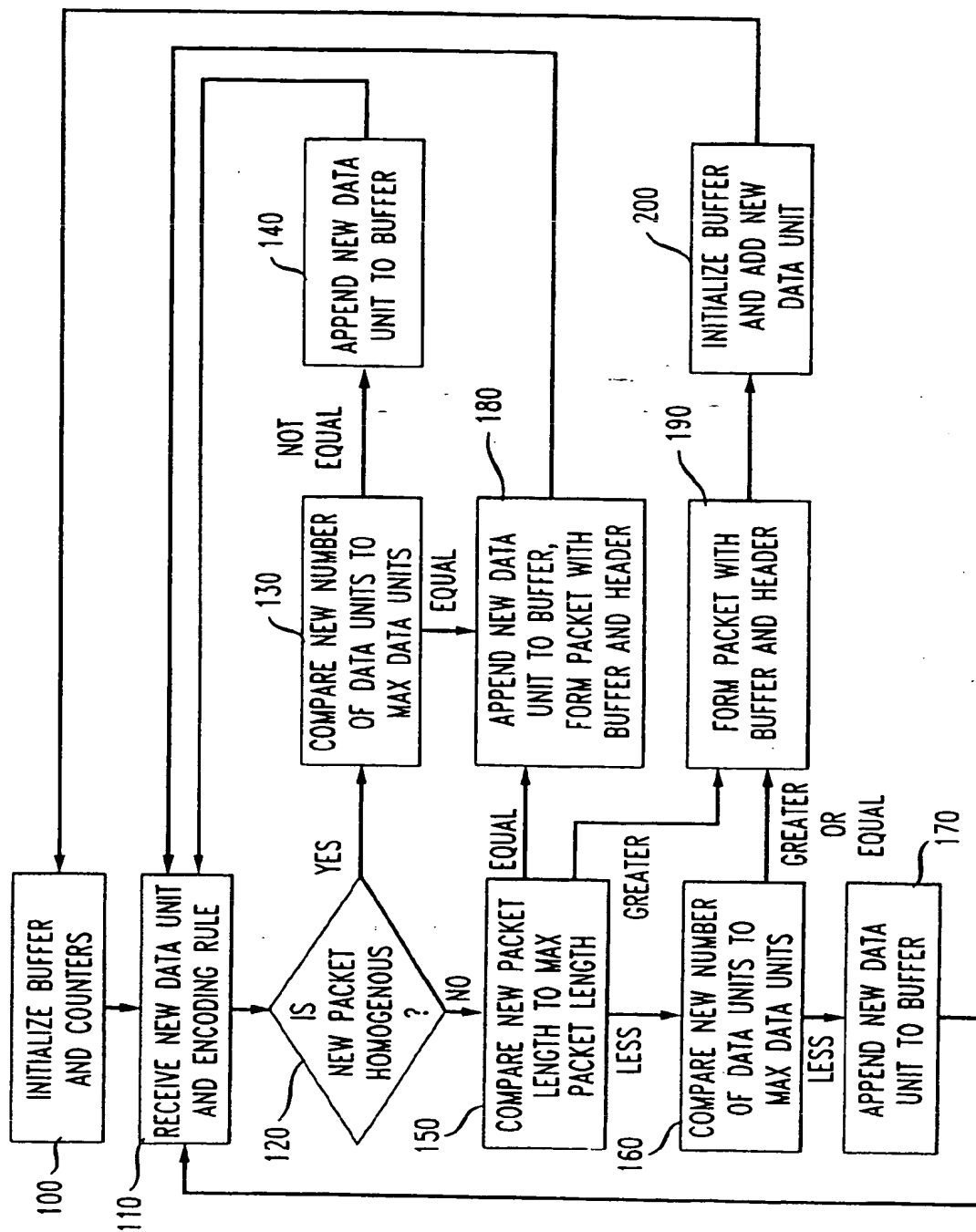24. The method of claim 23, wherein the packet is an AAL packet.

FIG. 1

*FIG. 2*

THIS PAGE BLANK (USPTO)